Guiding Performance optimization using the Roofline Model on Proxy Applications

Kristin Mackowski[1,2]

Advisor: Bob Robey[2] and Hai Ah Nam[2]

(1) New Mexico Institute of Mining and Technology, Socorro, NM 87801, (2) Los Alamos National Laboratory, Los Alamos, NM 87544

Abstract

Optimizing application performance should not only be guided by runtime identification of bottlenecks, but also the limitations imposed by different hardware components. Benchmarking proxy applications is important to understand where there is opportunity to improve performance in full application. In determining an application's potential for performance improvement, we identify an application's Arithmetic Intensity and plot this result on the Roofline Model[1] to map out the application's performance space. We invegistate the Arithmetic Intensity for the Stream Benchmark[2] and CLAMR[3] on different hardware architectures using the Roofline Model.

Background

- Application performance optimization typically involves identifying bottlenecks that consume the most run time. However, a timing analysis provides a limited picture of where and how much a code can be optimized.
- Efficiently using next-generation hardware, such as Trinity, with many-core Intel Xeon Phi, high-bandwidth memory, and DDR memory requires applications to exploit each hardware feature.

**Integrating hardware "bounds" to an applications performance analysis, to identify whether it is compute bound or memory bound is essential to performance optimizations on new architectures.*

Roofline Model

- The Roofline Model is a performance analysis tool that is used to benchmark hardware
- This model provides a graphical representation of the performance and scalability of multicore architectures.
- Application performance can be represented on the roofline model of the hardware to identify opportunities for performance optimizations, either focusing on compute or memory
- The roofline model involves measuring Arithmetic Intensity and GFLOPs per second. This model is the ratio of the peak GFLOPs/s vesas the stream bandwidth.
- Arithmetic Intensity is a measure of floating-point operations (FLOPs) performed by a given code relative to the amount of memory accesses (Bytes) that are required to support those operations.

Proxy Applications

•Proxy applications are a small-scale representation of a larger applications behavior. Analysis of proxy applications can provide insight and help guide optimization in the full application.

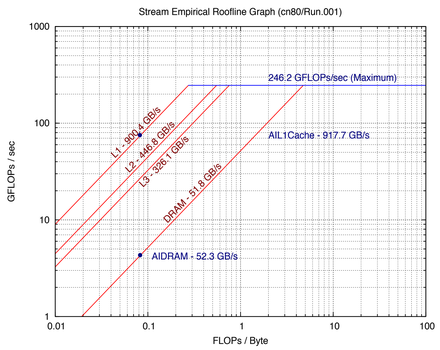
**STREAM is a benchmark that measures the Arithmetic Intensity of a application.*

**CLAMR is a Cell-Based Adaptive Mesh Refinement proxy application that was developed as a testbed for hybrid algorithum developement. The MPI only portion of CLAMR is being used in this project*.

Results

Sandybridge

Intel Xeon e5-2660_0; 8 cores

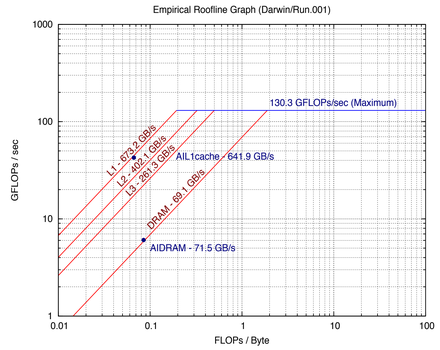


Graph 1: Stream 4 ranks on Sandybridge Node

Graph 4: CLAMR 4 ranks on Sandybridge Node

Haswell

Intel Xeon e5-2650_v3; 14 cores

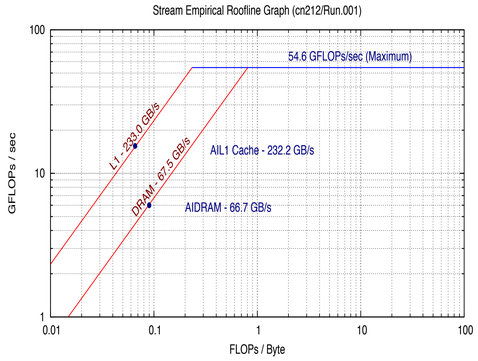


Graph 2: Stream 4 ranks on Haswell Node

Graph 5: CLAMR 4 ranks on Haswell Node

KNL

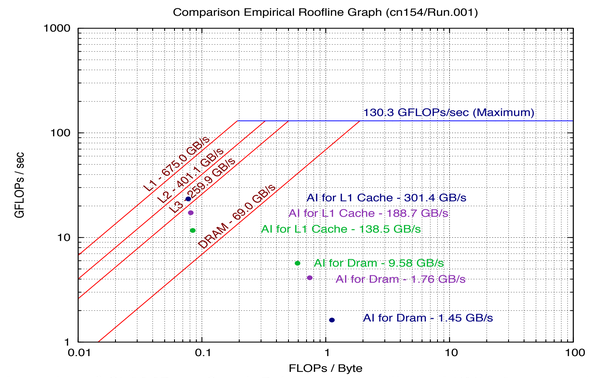
Intel Xeon Phi 7210; 14 cores



Graph 3: Stream 4 ranks on KNL Node

Graph 6: CLAMR 4 ranks on KNL Node

Analysis



Combined results of CLAMR on SandyBridge, Haswell, and KNL. As seen above, the gap between the L1 Cache and the DRAM is becoming smaller for each hardware. This shows that you are getting closer to your applications peak performance. In this graph you find that the KNL has the smallest gap and the best performance. The green is the KNL, the purple is Haswell and the blue is the Sandybridge.

Conclusion

Next-generation architecture has many new memory and compute components. To exploit the benefits you need to understand the hardware limitations and how your application relates to these constraints. Performance of different applications must integrate the Roofline Model to guide the code optimization. As seen above hardware impacts application performance. With CLAMR the newer the hardware the bigger the impact of performance optimization.

Acknowledgments

[1] Roofline Model: The Roofline Model is a hardware benchmarking tool. You can find this at https://bitbucket.org/berkeleylab/cs-roofline-toolkit .

[2] Stream Benchmark: This is a analysis tool that calculates the Arithmetic Intensity for each Proxy Application. You can find this at http://www.nersc.gov/users/application-performance/measuring-arithmetic-intensity/.

[3] CLAMR: CLAMR is a Cell-Based Adaptive Mesh Refinement proxy application. It is written in C++, MPI, OpenMP, and OpenCL. You can find this application at https://github.com/losalamos/CLAMR.

[4] I also would like to include all of the people who helped me with this project. Yuliana Zamora, Peter Ahrens, Priscilla Kelly, and Justin Litez.